(2)

Report No: CG-ONSCEN-04-90

SYNC2 PROGRAM ASSESSMENT REPORT

SYNETICS
540 Edgewater Drive
Wakefield, MA 01880

12 July 1990

FINAL TECHNICAL REPORT

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
United States Coast Guard

USCG Omega Navigation System Center
Alexandria, VA 22310-3998

Report No:

SYNC2 PROGRAM ASSESSMENT REPORT

SYNETICS
540 Edgewater Drive
Wakefield, MA 01880

12 July 1990

FINAL TECHNICAL REPORT

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
United States Coast Guard

USCG Omega Navigation System Center
Alexandria, VA 22310-3998

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

SYNC2 Program Assessment Report

5. Report Date 12 July 1990

6. Performing Organization Code SYNETICS Corp.

7. Author(s)

T.A. Palka, G.L. Noseworthy, W.F. O'Halloran, S. Eliassen

8. Performing Organization Report No.

9. Performing Organization Name and Address

SYNETICS Corp.
540 Edgewater Drive
Wakefield, MA 01880

10. Work Unit No. (TRAIS) J491-11

11. Contract or Grant No. DTCG23-86-A-20022

12. Sponsoring Agency Name and Address

Department of Transportation
U.S. Coast Guard
Omega Navigation System Center
Alexandria, VA 22310-3998

13. Type of Report and Period Covered

Final Report
March 1990 to July 1990

14. Sponsoring Agency Code USCG/ONSCEN

15. Supplementary Notes

16. Abstract

This report provides a technical assessment of the SYNC2 program currently used in the Omega system synchronization process. The utility of Omega depends upon the predictability of the phase and phase difference contours near the earth's surface. To achieve this predictability, a close synchronization of signals from all Omega transmitters is necessary. The SYNC2 program was developed in the mid-1970's to improve this synchronization process, and replaced an earlier synchronization program. From a performance viewpoint, the SYNC2 program has served its original purpose. Technological developments and improvements in estimation techniques, however, have rendered the program and accompanying documentation unsuitable as a baseline for further development or even continued use with the new Global Positioning System (GPS) operating environment. Many of the shortcomings with SYNC2 may be directly related to historical factors. SYNC2 was developed prior to the widespread acceptance of structured programming methods. Because of this, the SYNC2 software is extremely difficult to troubleshoot and to upgrade. SYNC2 was also developed before long term performance histories were available that would have facilitated SYNC2 algorithm development. The approach used by SYNC2 to calculate key algorithm parameters is cumbersome and could be significantly streamlined. The aggregate effect of such factors is that SYNC2 represents an overly complex, cumbersome, anachronistic mix of sometimes contradictory processes, routines and algorithms that no longer serves the Omega community's needs to provide reliable synchronization of the Omega System. The report examines three major aspects of the SYNC2 program: Functional Design, Software Implementation, and Software Documentation. Specific shortcomings in each of these three broad areas are examined. As a result of this assessment, a path to upgrading SYNC2 using modern programming techniques and estimation practices is set forth. (KR) (T

17. Key Words	18. Distribution Statement
Omega, GPS, Synchronization, Loran-C, Kalman Filter	

19. Security Classif. (of this report)	20. SECURITY CLASSIF. (of this page)	21. No. of Pages	22. Price
unclassified	unclassified		

Form DOT F 1700.7 (8/72) Reproduction of form and completed page is authorized.

TABLE OF CONTENTS

PAGE NO.

LIST OF FIGURES

PAGE NO.

LIST OF TABLES

PAGE NO.

1. INTRODUCTION

1.1 SCOPE AND PURPOSE

The SYNC2 program, developed in the mid 1970's, is currently used in the Omega system synchronization process. The purpose of this assessment report is to identify technical and programming deficiencies in the current SYNC2 version and to identify options to correct these deficiencies.

1.2 OMEGA OVERVIEW

The utility of Omega as a radionavigation system depends upon the predictability of the phase and phase difference contours near the earth's surface. To achieve this predictability, a close synchronization of signals from all the Omega transmitters is necessary. At each Omega station a set of cesium frequency standards is used to control the time of transmission of the signal. Differences between the frequency standards at each station result in timing (or phase) offsets that steadily increase if no control is exercised. The relative timing offsets between individual Omega transmitters are referred to as internal synchronization offsets.

Although Omega could operate without reference to any outside time standard, it is desirable to relate the Omega epoch to a fixed time standard such as Universal Coordinated Time (UTC). The average over time of the offset relative to an external reference such as UTC (U.S. Naval Observatory (USNO)) is referred to as an external synchronization offset. To prevent uncontrolled time offset growth in the Omega system, internal synchronization is accomplished by periodically adjusting the epoch of each transmitter to Mean Omega system time. External synchronization, which is not necessary for navigation, but is required for time dissemination, can be established by maintaining Mean Omega system time at a known constant offset from UTC.

A conceptual diagram of the Omega synchronization process is presented in Figure 1-1. Timing corrections are computed weekly at the Omega Synchronization Control Center using both internal and external measurements. Internal measurements are derived from reciprocal VLF phase difference data recorded by the monitor stations associated with the Omega transmitters.

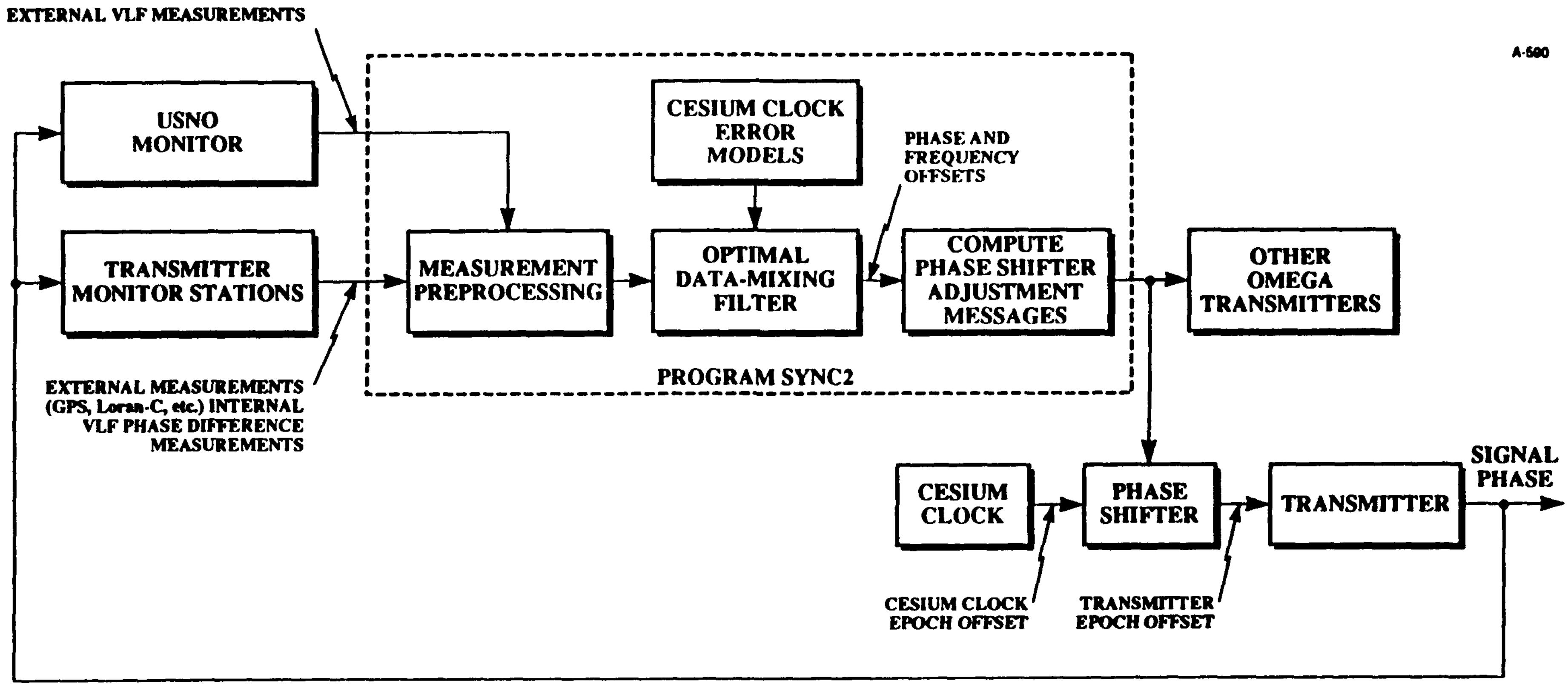

Figure 1-1 Omega Synchronization Process

These measurements are formed once each day. External measurements, which indicate the time offset between individual Omega transmitters and UTC, are provided by independent time sources such as VLF Phase, Loran-C, portable clock and the Global Positioning System (GPS).

A major element in the synchronization process is the SYNC2 program. On Monday of each week, each station compiles a SYNC DATA message containing the daily phase difference readings for each path as well as any available external measurements for each station for the last eight days (the eight day interval provides for a one-day overlap). These SYNC DATA messages are sent to Japan's Maritime Safety Agency (JMSA) and Omega Navigation System Center (ONSCEN) where the data are entered into the SYNC2 program. Central to SYNC2 is a Kalman filter which processes the many internal phase difference measurements and external measurements to estimate the internal offset of each station from the mean time of all stations, the estimated frequency offset of each cesium standard, and the estimated offset of Omega epoch from UTC. A principal output of the program is the estimated time offset from mean system time of each station at 0600 and 1800 UT for each of the last eight days. A weekly adjustment, or correction (CORR), is calculated as the amount of phase shift to be applied by each station to correct the estimated offset from mean system time. An accumulative phase adjustment (ACCUM) is also computed for each cesium standard. The ACCUM value, inserted every day at four hour intervals, compensates for the estimated rate of drift of each cesium. The CORR and ACCUM phase adjustments are sent to each station as a SYNC DIRECTIVE.

1.3 SYNC2 HISTORICAL PERSPECTIVE

Although the concept of Omega dates back to the late 1940's, the Navy did not establish the system until 1965. The first Omega station developed under this program became operational in 1972, even though experimental stations had been established in the late 1950's. Theoretical aspects of the synchronization problem were examined by the Navy early on and the reported results of these efforts lead to the first operational schemes. In 1971, the Coast Guard joined the Omega team by forming the Omega Navigation Systems Operation Detail (ONSOD). Although a Coast Guard Unit, ONSOD remained under operational control of the Navy until 1978. While the Coast Guard had had operational experience with radionavigation, they had no prior experience with Omega. Therefore, in such matters as synchronization methods, the "operational" control was directly influenced by Navy personnel.

To improve the synchronization procedure, development of the SYNC2 program was undertaken in 1974. This program, developed by a contractor under ONSOD direction, replaced an earlier synchronization program which was executed on a Wang calculator. The initial version of SYNC2 was released in January 1975 followed closely by SYNC2, version 2, released in April 1976. In the next several years, both the ONSOD and the JMSA gained considerable operational experience with SYNC2. During this operational period, several problems with SYNC2 were identified. A number of these problems were found to be the result of operator errors caused, in part, by the lack of complete information regarding SYNC2 operation. Design problems were also identified. Problems resulting from software design limitations fell into three principal categories:

- Poor quality measurement data input to SYNC2 was not automatically rejected by the program

- Extended transmitting station off-air periods sometimes caused SYNC2 to compute poor (erratic) synchronization adjustments

- Large bias errors in USNO measurements were not corrected by SYNC2.

To address these problems, a set of software modifications were implemented and released in November 1979 in SYNC2 version 3. It should be noted that not all the problems identified during this period resulted in software modifications. Of the total of 11 problem reports submitted, 3 were deferred until further data became available and 2 were identified as being outside the scope of the effort [Reference 6 (Appendix B)].

At the time SYNC2 version 3 was released, the SYNC2 contractor also made recommendations for further program modifications. Three specific items were identified:

- Additional Omega signal phase measurements taken at Omega signal frequencies 11.05 and 11-1/3 kHz

- Modification of the Kalman filter implementation to optimally estimate phase measurement bias errors

- Measurements of station clock error relative to UTC taken with the GPS.

These proposed modifications were intended to improve synchronization under anomalous conditions by improving identification and removal of measurement bias errors in Omega phase measurements and reducing the requirement for operator intervention. Only the last of these modifications has been fully implemented in the current SYNC2 version.

Regarding the first proposed modification, the present monitor station receivers do not make measurements at 11.05 kHz and therefore such internal VLF measurements are not available for SYNC2 processing. The 11-1/3 kHz frequency measurements are available and may be of some marginal interest, but their utility is probably not sufficient to balance the additional data collection requirements associated with their use.

Regarding the second proposed modification, SYNC2 version 3 implemented a scheme to account for bias errors which can appear in external Omega VLF phase measurements (i.e., those measurements relative to UTC recorded at USNO). To compensate such biases, SYNC2 was designed to compute a bias correction factor using other external measurements such as Loran-C. This correction factor allows any biases to be removed from the external VLF measurements prior to SYNC2 processing. In SYNC2 version 3, this correction factor was computed via an averaging processing [Reference 6, Appendix D]. The second modification listed above proposed to improve the bias error calibration process by using an optimal estimator. This modification has not been implemented, and given the diminished importance of external VLF measurements, the benefits of moving to an optimal estimator for the correction factor are negligible. Finally, we note that in the main SYNC2 software documents [References 5 and 6], this bias correction factor is referred to as "CORR". The SYNC2 user community, however, currently uses the term "CORR" to refer to the weekly phase adjustments that are sent as a SYNC DIRECTIVE.

It is also useful to examine the state of estimation practices during the early 1970's and other factors that influenced the original SYNC2 design. The baseline development of SYNC2 was heavily influenced by the then emerging field of Kalman filtering. Kalman filter applications were a specialty of the SYNC2 contractor so it was natural that the Kalman filter was to become the centerpiece of the SYNC2 program. There was little incentive at the time to find good engineering approximations to the synchronization problem. The brute force approach using available matrix routines and complex measurement differencing schemes could be applied wholesale with less

technical risk than looking for simpler alternatives. The Kalman filter implementation approach used in SYNC2 preceeded the development of efficient, numerically stable algorithms that have become the standard in today's Kalman filter implementations.

A second factor impacting the baseline SYNC2 development was lack of long term performance histories that would have facilitated the SYNC2 measurement model development. This lack of data resulted in a development approach that relied heavily on a History File containing several weeks' worth of recent raw data to specify parameters defining the model, rather than relying on fixed, empirically verified values for key filter parameters.

Also, because SYNC2 was developed before the advent of GPS, or the widespread availability of Loran-C, the program design focused on the utilization of internal reciprocal path VLF measurements. A significant portion of the program is devoted to the editing, propagation correction calculation and massaging of these internal measurements. The ability to process external measurements was an "add-on" feature. This also helps explain the major program effort devoted to the pre-processing of internal VLF measurements. In contrast, the external measurements are treated in a more straightforward manner.

Finally, the coding of SYNC2 itself was performed before the popularization of modular, top-down, self documenting structured programming techniques which resulted in a program which is extremely difficult to troubleshoot, modify or understand. Recent experiences, related mainly to the utilization of GPS data, have highlighted the need for program and documentation improvements. Questions posed to ONSCEN by the JMSA also showed that the current documentation fails to provide a complete picture of the SYNC2 fundamentals regarding GPS data, which is becoming increasingly important relative to the internal reciprocal phase measurements as a source of both internal and external synchronization estimates.

The cumulative effect of the factors described above is that SYNC2 represents an overly complex, cumbersome, anachronistic jumble of sometimes contradictory processes, routines and algorithms that no longer serves the Omega community's needs to provide reliable synchronization of the Omega System. It is therefore timely to provide this assessment of the SYNC2 technical and programming deficiencies, and identify a set of corrective actions.

1.4 REPORT ORGANIZATION

An evaluation of the current SYNC2 program is provided in section 2. For convenience the assessment has been divided into three main segments:

- Technical/Algorithmic Assessment (Section 2.2)
- Software Assessment (Section 2.3)
- Documentation Assessment (Section 2.4).

Within each of these sections individual topics are examined separately. Based on this assessment, conclusions and recommendations are presented in section 3, and two distinct options for enhancing SYNC2 are examined.

2. SYNC2 PROGRAM ASSESSMENT

The assessment of the SYNC2 program is divided into several sections. Following a brief program overview presented in section 2.1, the technical aspects of the program are assessed in Section 2.2. This assessment section examines the data processing techniques employed in SYNC2, from a technical (as opposed to software) viewpoint. Section 2.3 assesses the program implementation and notes programming deficiencies. Finally, the state of the SYNC2 software documentation is examined in Section 2.4.

2.1 PROGRAM OVERVIEW

Program SYNC2 mechanizes an integrated dynamic synchronization process in the form of a data-mixing filter that includes:

- A mathematical model for cesium clock frequency and phase offset dynamics for each station

- A quantitative representation of the statistical uncertainty in the internal and external measurement inputs.

The SYNC2 program consists of a main program and 27 subroutines written principally with FORTRAN IV statements. The general structure of the program is shown in Figure 2.1-1 and the calling structure is shown in Figure 2.1-2. The program performs three main functions: computes both internal and external synchronization offsets, as well as generates phase shifter adjustment messages for each Omega station.

The largest of the input sets illustrated in Figure 2.1-1 is the History File, containing SYNC2 input and output data from previous weeks. Information from the History File is used for several purposes including computing measurement statistics for error rejection and measurement quality weighting. Also of interest is the PPC (Predicted Propagation Corrections) file containing the PPC's for all internal and external propagation paths. SYNC2 applies the PPCs and charted nominal phase shifts to the VLF timing data to obtain synchronization offset measurements.

PROGRAM SYNC2 – GENERAL STRUCTURE

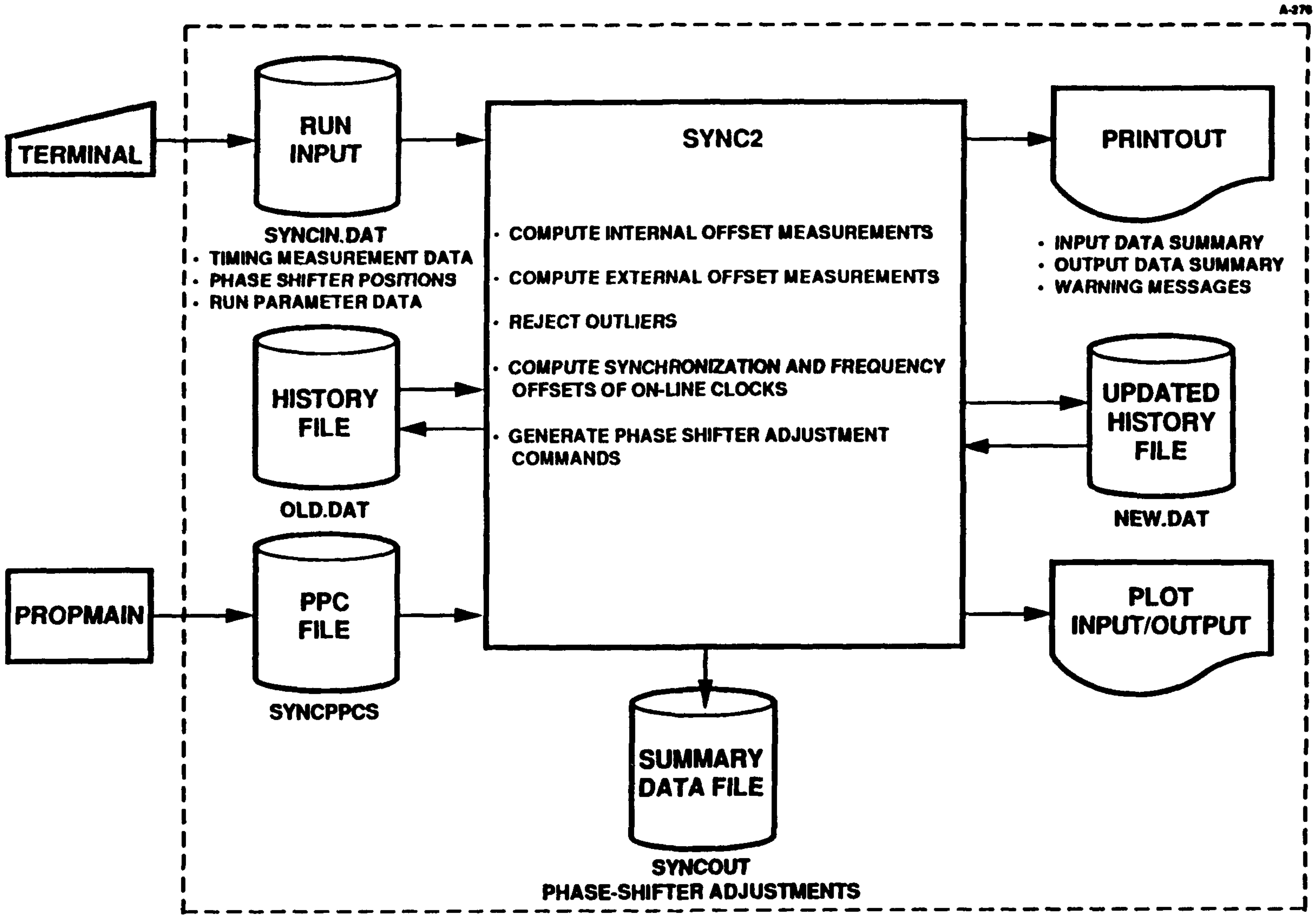

Figure 2.1-1 General Structure of SYNC2

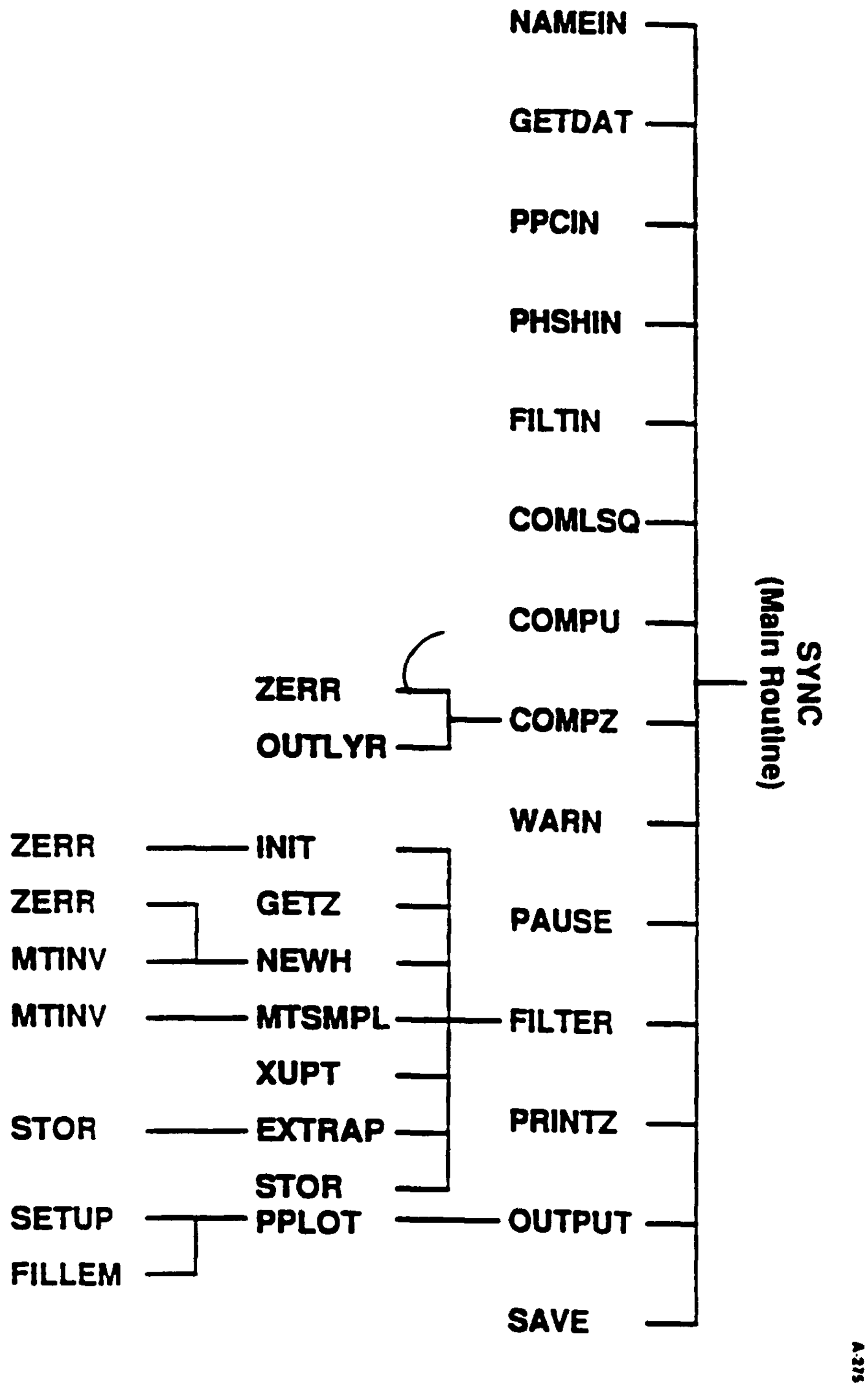

Figure 2.1-2 SYNC2 Routine Calling Structure

2.2 TECHNICAL / ALGORITHMIC SHORTCOMINGS

From a performance viewpoint, the SYNC2 program has served its original purpose; however, technological developments and improvements in estimation techniques have rendered the program and accompanying documentation unsuitable as a baseline for further development or even continued use with the new GPS operating environment. In the following sections several design shortcomings are examined. Section 2.2.5 presents some performance problems observed in SYNC2 operations.

2.2.1 Kalman Filter Algorithm

The SYNC2 implements a batch mode Kalman filter. Although at the time of development this approach was state of the art, implementations with improved numerical properties, such as Biermann's sequential U-D algorithm [Reference 8], have since become the standard in estimation applications. For example, the navigation filter in the DoD GPS receiver's is based on Biermann's algorithm.

Sequential U-D (or square root) algorithms have several advantages over the standard batch mode Kalman filter algorithms, such as the one implemented in SYNC2. Batch measurement processing requires matrix inversion and in SYNC2 the required matrices may be larger than 16 x 16. This is perhaps the most obviously wasteful and numerically suspect aspect of the SYNC2 filter algorithm. Sequential processing of measurements, whether based on covariance or factorization schemes, avoid the need for matrix inversion and reduce the storage requirements.

Experience in various estimation applications have shown that the standard Kalman filter algorithm is sensitive to computer roundoff and that numerical accuracy sometimes degrades to the point where the results cease to be meaningful. Numerical problems may be particularly acute in situations with near perfect measurements or near singular covariance matrices. One manifestation of numerical problems is divergence, i.e., a situation where the actual errors grow in an unbounded fashion. The effects of numerical errors are generally manifested in the appearance of covariance matrices that fail to retain nonnegativity (i.e., nonnegative eigenvalues). Several different *ad hoc* methods are often used to combat numerical problems. One method used in SYNC2 is to force

the filter-computed covariance matrix to be symmetric by averaging appropriate off-diagonal matrix elements.

Distinct from such *ad hoc* methods are methods based on factorization techniques which yield algorithms with inherent stability and better numerical accuracy when compared to standard Kalman filter algorithms. The improved numerical behavior of square root (or U-D) algorithms is due in large part to a reduction in the numerical range of the variables. Loosely speaking, one can say that computations which involve numbers ranging between 10^{-N} and 10^{N} are reduced to ranges between $10^{-N/2}$ and $10^{N/2}$. Thus the square root algorithms achieve accuracies that are comparable with standard Kalman filters that use twice their numerical precision.

2.2.2 Data Derived R-Matrix

The measurement noise covariance matrix R sets the uncertainty associated with each measurement and models any correlations that may exist between the measurement errors. By way of the Kalman gain matrix, the measurement noise matrix determines the relative weighting of each measurement in forming the state estimate. The performance of the filter may be degraded if the R matrix does not properly model the actual measurement error statistics.

At the time of the SYNC2 development, the Omega system was still in the developmental stage and without a long-term performance history. Program parameters such as the measurement noise variances were heavily dependent on history files rather than nominal values based on representative statistics. Specifically, measurement variances for the internal VLF phase measurements are computed within SYNC2 using 3 weeks' worth of measurement residuals. This approach, prudent in the developmental time frame, explains the massive data dependencies that appear throughout the program. There is no theoretical foundation for this cumbersome method and, given the performance histories that now exist, the original justification for this approach no longer exists. An improved approach would make use of nominal baseline noise values with the measurement residuals used as reasonability checks or as augmentation factors. These nominal values would then be derived, offline, from the available historical data. Because of seasonal variations, a single constant noise value may not provide a sufficiently accurate representation of the error characteristics of the process. To overcome this limitation, a set of nominal noise values for appropriately defined time intervals can be derived. As a practical matter, this approach may be difficult to implement since the existing historical data is not in a form that may be easily processed.

In contrast with the complex scheme implemented for VLF measurements, fixed nominal values are used to construct the R-matrix components associated with the non-VLF external measurements. These external measurements, specifically GPS, are now assuming a greater role in the synchronization process. Accordingly, an improved approach for handling the expected uncertainty in these measurements would be to use nominal baseline noise values based on available historical data, with measurement residuals used as reasonability checks or augmentation factors.

2.2.3 Measurement Differences/Processing Schedules

Because of temporal correlation of the internal phase measurement errors (due to propagation effects), SYNC2 implements a complex scheme to assure uncorrelated measurement errors. A much simpler, alternate approach would be to prefilter measurements and update the filter with the derived uncorrelated measurements at a much slower rate (e.g., two times a week).

The feasibility of performing the measurement update step of the filter at a slower rate depends in large part on the frequency stability of the cesium frequency standards. A recent study [Reference 7] suggests that the cesium frequency standards are sufficiently stable that no noticeable differences would occur if the interval between measurement updates were to be extended beyond the nominal interval currently set at one day (at least during steady state). This would greatly simplify the synchronization process. This item would require additional performance studies to verify the suggested approach.

2.2.4 External Measurement Sources: GPS and Loran-C

At the time SYNC2 was designed, the set of internal reciprocal path phase measurements represented the primary measurement source. External measurements, such as those provided by a portable clock, were also available, but only at relatively infrequent intervals. This situation influenced the signal performance specification, and hence, the development of SYNC2. SYNC2 was developed to maintain internal synchronization, that is synchronization of each transmitter to a common time, termed Omega System Time, using the internal VLF phase measurements. External synchronization -- synchronization of Omega system time to UTC -- was to be maintained using an external measurement source such as a portable clock. The design of SYNC2, however,

does not preclude the use of solely external measurements to establish both internal and external synchronization.

The availability of Loran-C data at Norway, Hawaii, North Dakota and Japan and the advent of GPS now provide a source of daily external measurements. This change in the relative availability of external measurements has had a profound impact on the synchronization process. These external sources allow a means to tie Omega time to UTC time on a near real time basis (as well as obviating the need for a portable clock visit in case of synchronization loss).

The deployment of GPS receivers at Omega stations began in 1985 at the Liberia station, and by 1987 all stations had access to this source of highly precise timing information. These GPS receivers were initially deployed for the purpose of avoiding expensive losses in system availability if a failure should cause loss of coarse synchronization. This data, however, was also available for synchronization purposes and by 1989 the transition from Loran-C to GPS at the northern stations was complete.

The utilization of GPS measurement data has a significant impact on SYNC2 performance and on the overall synchronization process. In practice, the utilization of GPS data at each station has allowed each station to synchronize directly to UTC and the distinction between internal and external synchronization has virtually disappeared. The strong influence exerted by the GPS measurements stems from their relative weighting in the SYNC2 filter. While the complex data-dependent model for the internal VLF measurements yields RMS accuracy values (used in R matrix) of between 0.5 μsec and 2.5 μsec, the RMS accuracies of the GPS measurements are apparently fixed in SYNC2 at 0.06 μsec. This weighting scheme accounts for the strong influence GPS data exerts on the SYNC2 outputs.

Some insight into the role of external GPS measurements and internal VLF measurement may be gained from an examination of actual test data. Table 2.2-1 provides a comparison of two SYNC2 runs over the same data set. In the first run the weekly adjustments were computed by SYNC2 on the basis of both daily GPS phase difference measurements and internal reciprocal path VLF measurements. For comparison purposes, a second run was performed with the internal

TABLE 2.2-1
SYNC2 Performance Comparison

SYNC2 VERSION 02
OUTPUT DATA SUMMARY
PHASE SHIFTER ADJUSTMENTS

TRANSMITTER	CESIUM STD.NO.	STATUS	WEEKLY ADJUSTMENT (MICROSECONDS) GPS/PHASE DIFF	WEEKLY ADJUSTMENT (MICROSECONDS) GPS INPUT ONLY
A	212400	ONLINE	.12	.08
A	201701	1ST BACKUP	.12	
A	148600	2ND BACKUP	.12	
B	125601	ONLINE	-.15	-.14
B	165500	1ST BACKUP	-.15	
B	162300	2ND BACKUP	-.15	
C	152900	ONLINE	-.03	-.03
C	155400	1ST BACKUP	-.03	
C	134900	2ND BACKUP	-.03	
D	139700	ONLINE	-.35	-.38
D	202800	1ST BACKUP	-.35	
D	208500	2ND BACKUP	-.35	
E	132001	ONLINE	-.44	-.45
E	109100	1ST BACKUP	-.44	
E	52200	2ND BACKUP	-.44	
F	137100	ONLINE	-.18	-.17
F	131801	1ST BACKUP	-.18	
F	42600	2ND BACKUP	-.18	
G	131900	ONLINE	-.16	-.14
G	126803	1ST BACKUP	-.16	
G	155300	2ND BACKUP	-.16	
H	167102	ONLINE	-.26	-.27
H	122401	1ST BACKUP	-.26	
H	221600	2ND BACKUP	-.26	

measurements turned off so that only daily GPS measurements were available. A comparison of the SYNC2 outputs for these two runs shows very little difference in the computed adjustments. The maximum difference between the computed adjustments for this example is 0.04 μsec with an RMS error at 0.01 μsec.

The evident importance of the GPS data on the SYNC2 outputs raises several issues. First, since the incorporation of internal measurements requires lengthy data inputs (512 measurements) by the SYNC2 operator and since their effect on the SYNC2 outputs is negligible, it might be desirable to omit internal measurements entirely when GPS data is available. This change would result in profound simplifications to the SYNC2 program without loss of performance. SYNC2 must, however, retain the ability to process internal measurements, since GPS is still experimental and may not always be available.

Second, given that the GPS plays such an important role in SYNC2 processing, the measurement model must be carefully verified and documented. Unlike the internal and external VLF measurements, the error variances associated in SYNC2 with non-VLF measurements are fixed values. As mentioned early, SYNC2 apparently uses a value of 0.06 μsec RMS for the GPS measurement components of the R matrix connect.

This value is consistent with performance specifications for the GPS control and space segments, but empirical data should be used for verification. In addition to re-examining the baseline measurement noise value used in SYNC2, the possibility of including some adaptivity to account for GPS space/control segment anomalies and errors in ionospheric compensation (due to solar activity), should be considered. One means to include this adaptive feature is to make use of timing accuracy indications that may be available from the onsite GPS receivers ; a second option would be to make use of the satellite User Equivalent Range Error (UERE) available from the GPS bulletin board provided by Yuma.

2.2.5 Predicted Propagation Corrections

The PPC input file, one of three input files used by SYNC2, contains the Predicted Propagation Corrections (PPCs) generated by a separate program. The PPCs are designed to account for VLF propagation delays resulting Earth's magnetic field and other non geometric factors, and are very slightly direction dependent. The process by which the PPCs are incorporated into SYNC2 is illustrated in Figure 2.2-1.

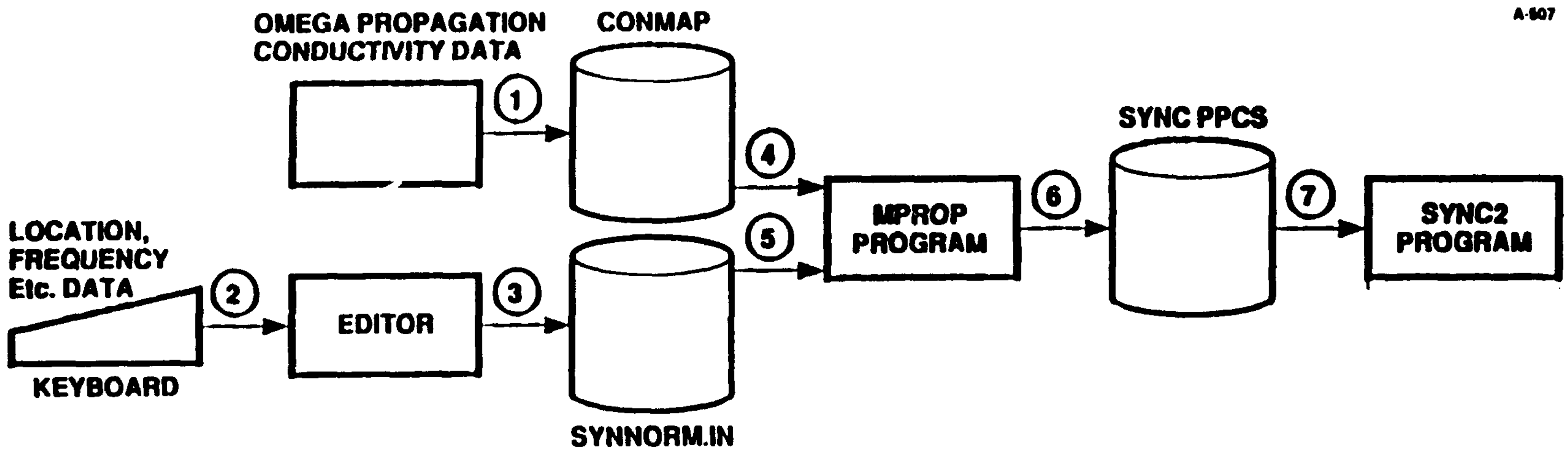

Figure 2.2-1 PPC File Generation

Given the performance histories that now exist, it should be possible to compute corrections based on actual measurement data rather than relying on the PPCs. In any event, the role played by the PPCs in the processing of internal VLF measurements is negligible; it has not been established, for example, that the employed PCCs display any significant reciprocity variations. All PPC errors caused by direction dependent anomalies (e.g., solar illumination and ground conductivity) tend to cancel out in the differencing process used for the internal reciprocal path measurements.

2.2.6 System Transitions

The one area in which apparent SYNC2 filter performance problems have been noted is in the transition between different operating configurations. System transition include situations such as:

- When a station has been off-air for a considerable period of time and then resumes signal transmission
- When a portable clock measurement shows an error which is much larger than expected
- When an external GPS measurement is removed from a station.

The SYNC2 program covariance matrix contains uncertainties in station clock error estimates and the correlations between errors in the estimates. This covariance matrix changes with each new measurement processed by the SYNC2 Kalman filter algorithm which modifies the matrix to reflect the situation at the new measurement time. During SYNC2 version 2 testing, it was observed that under the first two circumstances cited above, the covariance matrix was not reflective of the actual error statistics. In SYNC2 version 2, the correlations between the off air station clock and the other Omega stations were maintained by the SYNC2 filter. The greater these correlations, the greater the chance that all stations estimates will be strongly affected and behave erratically when the off-air station resumes transmissions and its Omega phase measurements begin to be processed in SYNC2 again.

To prevent this erratic behavior, both the uncertainties in the estimates and the correlations between errors in the estimates must be adjusted to reflect the true behavior of the off-air station. SYNC2 version 2 was modified (yielding version 3) so that this adjustment in the covariance matrix

took place automatically. Although this modification apparently yielded some improvement in performance, not all cases were adequately addressed, and operator intervention may be required in some off-air situations [Reference 6].

More recently [References 1, 2], performance problems have been noted when external timing is applied and then removed from a station. When external timing is applied to a station, it often shifts two to three microseconds from its internally referenced time. When the external measurements are removed, however, the program tries to drive that station back to its internally referenced time, which is often two or more microseconds "in error" from its externally referenced state. This behavior is due, in part, to an imperfect understanding of propagation which is the key to determining the internal synchronization offsets. To investigate the described behavior in more detail, several different scenarios were considered using a simulation of the major elements of SYNC2. The results of these runs indicate that if the internal and external measurements are mutually consistent, the SYNC2 outputs should remain stable when external measurements are removed. If, however, an inconsistency of sufficient magnitude exists between the internal and external measurements, the SYNC2 filter may exhibit the performance described [References 1 and 2]. The results of these analysis test runs are provided in Appendix B. Although these simulation runs provide some insight into the observed behavior, a further examination of the problem using the actual test data should be included in benchmark testing for SYNC2 redesign efforts.

2.3 Software Assessment

Viewed as a software item, SYNC2 is deficient in several respects. During the original development of SYNC2, structured programming techniques were not yet standard in the industry. In addition, SYNC2 was not developed to meet any MIL-STD software specification. As a partial result of these factors, the architecture of SYNC2 is hopelessly outdated, especially on data inputs, parameter inputs, modular design, use of commons, etc. Each of the following subsections discuss the different classes of software problems that have been noted. A listing of specific coding errors, and other problems noted in each of the 27 modules is provided in Appendix A.

2.3.1 Programming Style/Maintainability

The programming techniques used in developing SYNC2 yielded a program that is difficult to decipher. The most significant "stylistic" problems are:

- Lack of dated revision history information
- Lack of indentation
- Inadequate code comments
- Inadequate data item descriptions
- Inadequate handling of program constants.

Examples of several of these types of errors are evident in Figure 2.3-1 which shows the SYNC2 routine COMLSQ. Code written in such a style is difficult to understand and hence is difficult to maintain and upgrade. For contrast, the same portion of code has been rewritten in Figure 2.3-2 using indentation and more numerous comments. The rewritten version also adds a revision history section which is required for configuration control.

The absence of data item descriptions makes it difficult to establish a correspondence between algorithms documented in the software functional description [Reference 5] and the code. This problem is exacerbated by the scarcity of adequate code comments.

Program constants, such as the GPS measurement noise variance, are handled inefficiently. Rather than storing parameters in a data file to be read, many system parameters are defined directly in the subroutine NAMEIN. Modifying these parameters thus requires NAMEIN to be recompiled and the program relinked. This approach is clearly undesirable and limits the ability to assess the sensitivity of the filter changes in the system parameters. A preferable approach reads the system parameters from a separate file or table which could then be easily modified, avoiding the requirement for a new link each time a parameter needs to be changed.

2.3.2 Computational Accuracy/Run-Time Efficiency

Several features of SYNC2 are potential sources of roundoff induced errors. No use of double precision is made for critical calculations. This omission can lead to loss of accuracy for

```
C
C     PURPOSE
C       CONLSQ COMPUTES A LINEAR LEAST SQUARES FIT TO PAST WEEKS OF
C       PHASE SHIFTER POSITIONS AND SETS THE SLOPE AS THE FREQUENCY
C       OFFSET FOR EACH CLOCK
C
C     ARGUMENTS
C       *NONE*
C
C     SUBPROGRAMS CALLED
C       *NONE*
C
C
C*****************************************************************
C
      SUBROUTINE CONLSQ
C
C
C
      REAL CLKRAT,CLOCK
C
C
      COMMON /CLK/ KLKNO(10,4,8),CLKRAT(10,4,8),CLOCK(10,4,8),
     1  CONTRL(8,4),KWEEKS,MESS(10,6,50)
      COMMON NTRANS,KARWKS
      DIMENSION CLKPOS(10)
C
      DO 40 IT=1,NTRANS
      DO 35 IC=1,4
      KNOW=KLKNO(KWEEKS,IC,IT)
      IF(KNOW.LE.0) GO TO 40
      DO 20 II=1,KWEEKS
C
      IW=KWEEKS-II+1
      DO 10 IK=1,4
      IF(KNOW.EQ.KLKNO(IW,IK,IT)) GO TO 15
10    CONTINUE
C     CLOCK NOT FOUND AT THIS WEEK, TAKE L.SQ. ONLY THIS FAR BACK
      IF(II.GT.2) GO TO 25
      CONTRL(IT,IC)=CLKRAT(KWEEKS,IC,IT)
      GO TO 35
15    CLKPOS(II)=CLOCK(IW,IK,IT)
      DO 17 I=1,50
C     OMSFOG JUMP
      DO 17 IJ=1,II
      IF (MESS(IW,1,I).EQ.KNOW .AND. MESS(IW,6,I).EQ.-1000)
     1 CLKPOS(IJ)=CLKPOS(IJ)-MESS(IW,2,I)/100.
17    CONTINUE
      IF(II.EQ.1) GO TO 20
      IF(ABS(CLKPOS(II)-CLKPOS(II-1)).GT.50.) CLKPOS(II)=CLKPOS(II)-
     1  SIGN(1.0E2,CLKPOS(II)-CLKPOS(II-1))
20    CONTINUE
      II=KWEEKS+1
25    II=II-1
C     II WEEKS OF HISTORY OF THIS CLOCK WERE FOUND
C     PERFORM LEAST SQUARES
      S1=0
      S2=0.
      DO 30 J=1,II
      S1=S1+CLKPOS(J)
30    S2=S2+CLKPOS(J)*(II+1-J)
      SLOP=(S2*2./(II+1)-S1)/(II*(II-1)/6.)
C     CONVERT FROM WEEKS TO 4 HR PERIODS

      CONTRL(IT,IC)=SLOP/42.
35    CONTINUE
40    CONTINUE
      RETURN
      END
```

Figure 2.3-1 Sample Section of SYNC2 Code

```
C
C     PURPOSE
C       COMLSQ COMPUTES A LINEAR LEAST SQUARES FIT TO PAST WEEKS OF
C       PHASE SHIFTER POSITIONS AND SETS THE SLOPE AS THE FREQUENCY
C       OFFSET FOR EACH CLOCK
C
C     ARGUMENTS
C       *NONE*
C
C     SUBPROGRAMS CALLED
C       *NONE*
C
C
C*********************************************************************
C  AUDIT TRAIL:
C  DATE      INITIALS                  DESCRIPTION
C 5/10/90       OWE          Removed old card-id from cols 72 and on
C                            Added indents to DO loops for readability
C 5/ 5/90       OWE          Copied from current version
C                            Inserted Audit Trail section
C*********************************************************************
C
      SUBROUTINE COMLSQ
C
C
C
      REAL CLKRAT,CLOCK
C
C
      COMMON /CLK/ KLKNO(10,4,8),CLKRAT(10,4,8),CLOCK(10,4,8),
     1  CONTRL(8,4),KWEEKS,MESS(10,6,50)
      COMMON NTRANS,KARWKS
      DIMENSION CLKPOS(10)
C
      DO 40 IT=1,NTRANS
         DO 35 IC=1,4
            KNOW=KLKNO(KWEEKS,IC,IT)
            IF(KNOW.LE.0) GO TO 40
C
            DO 20 II=1,KWEEKS
               IW=KWEEKS-II+1
               DO 10 IK=1,4
                  IF(KNOW.EQ.KLKNO(IW,IK,IT)) GO TO 15
10             CONTINUE

C     CLOCK NOT FOUND AT THIS WEEK, TAKE LEAST SQ. ONLY THIS FAR BACK

               IF(II.GT.2) GO TO 25
               CONTRL(IT,IC)=CLKRAT(KWEEKS,IC,IT)
               GO TO 35
15             CLKPOS(II)=CLOCK(IW,IK,IT)

C     OMSFOG JUMP

               DO 17 I=1,50
                  DO 17 IJ=1,II
                  IF (MESS(IW,1,I).EQ.KNOW .AND. MESS(IW,6,I).EQ.-1000)
     1            CLKPOS(IJ)=CLKPOS(IJ)-MESS(IW,2,I)/100.
17             CONTINUE
               IF(II.EQ.1) GO TO 20
               IF(ABS(CLKPOS(II)-CLKPOS(II-1)).GT.50.) CLKPOS(II)=CLKPOS(II)-
     1         SIGN(1.0E2,CLKPOS(II)-CLKPOS(II-1))
20          II=KWEEKS+1
5           II=II-1

C     II WEEKS OF HISTORY OF THIS CLOCK WERE FOUND
C     PERFORM LEAST SQUARES

            S1=0
            S2=0.
            DO 30 J=1,II
               S1=S1+CLKPOS(J)
30          S2=S2+CLKPOS(J)*(II+1-J)

C     CONVERT FROM WEEKS TO 4 HR PERIODS

            SLOP=(S2*2./(II+1)-S1)/(II*(II-1)/6.)
            CONTRL(IT,IC)=SLOP/42.
35       CONTINUE
40    CONTINUE
      RETURN
      END
```

Figure 2.3-2 SYNC2 Code Sample Revision

critical filter calculations. Moving to square root or U-D filter implementation might eliminate the need for extra double precision, at least in the filter section of the code. In addition, many examples of mixed mode computations may be found which are also a source potential numerical problems.

More than 90% of the SYNC2 code is in FORTRAN IV. This is very inefficient for present FORTRAN 77 systems. Upgrading to FORTRAN 77 would also further simplify the code and make it more readable.

2.3.3 Input Data Process

The input file for SYNC2, SYNCIN.DAT, contains 33 separate input parameters as well as the new measurement data. This file takes the form of a rigidly ordered list in an ASCII format. The contents of this input file are summarized in Table 2.3-1. This table provides the name of each variable, a brief description, and some information on usage. As indicated, 6 of the inputs appear to be unused within the code. Included in this set are parameters PPC102 and PPC136, intended to allow known PPC errors in one-way phase measurements to be corrected on input. Also unused is SPOWER. This variable input was added to SYNC2 version 3 to allow the program to determine if a time period of reduced transmitting station radiated power will affect Omega phase measurement accuracy [Reference 6]. The features associated with these "unused" parameters may no longer exist. The preparation of the SYNCIN.DAT file from the raw inputs provided by the various stations is a task that requires a good deal of general Omega knowledge, as well hours of manual preparation and data reduction.

The subroutine which reads in the SYNCIN.DAT file, NAMEIN, has several shortcomings. The variable name for parameters appearing in columns 1 through 10 is never read or checked; only the data, starting in column 12 is read. This is a dangerous procedure in that parameters which are inadvertently switched by the operator will not be detected by NAMEIN. The handling of the parameters EXDAT and EXTDAT listed in Table 2.3-1 illustrates a another problem. These variables, for which operator inputs are required, provide the "true/false" indication as to whether the history file will be accessed. As the routine is constructed, EXDAT will overlay data read in for EXTDAT; the EXTDAT contents are stored without change in EXDAT.

TABLE 2.3-1

SYNCIN.DAT Variables

Variable Number	Variable Name	*Use	Brief Description
1	MEASUREMENT PAIRS		Station pairs (AB AD...etc.)
2	EXDAT		External data switch (History File)
3	KARWKS		Number of weeks of new input
4	ENDATE		Date of last day of input (Monday)
5	EXSYNC		External synchronization
6	ADJLIM	E	Omega synchronization limit in microseconds
7	PLOT		Plot enabling switch
8	SIGFSQ	E	Clock model rate squared
9	SIGFD2	E	Clock model frequency fluctuation squared
10	TAU	E	Measurement error time correlation
11	EXTDAT		External data switch (History File)
12	LEAPOF	E	Leap second adjustment UTC to Omega System Time

* Variable Use: "E" means engineering judgment is required for correct usage

"X" means that this variable input is discontinued, or program code prevents its use

A blank means that it is regular input and does not require special handling.

TABLE 2.3-1 (con't)

Variable Number	Variable Name	*Use	Brief Description
13	PPC102	X	Run time 10.2 kHz PPC input
14	PPC136	X	Run time 13.6 kHz PPC input
15	LATE		Late incorporation of corrections for station
16	SESCAL		Plot max y-scale for synch offset estimates
17	SUSCAL		Plot max y-scale for synch offset uncertainties
18	FESCAL		Plot max y-scale for frequency offset estimates
19	FUSCAL		Plot max y-scale for frequency offset uncertainties
20	TRACKT	X	Set tracking mode for transmitter
21	OMSFOG		OMSFOG fault adjustment input
22	STATUS		Clock status input
23	INITP	E	Initialization data for phase and frequency offset estimates
24	INITX	E	Initialization data for rms values of phase and frequency offset estimates
25	VLFERR	X	VLF phase data for USNO
26	NONERR	E	Overrides rms accuracy of external synch measurements

TABLE 2.3-1 (con't)

Variable Number	Variable Name	*Use	Brief Description
27	MESTIM		GMT input for portable clock measurement
28	DAYNIT		Day/night flags for control times
29	BIAS	X	External timing bias for USNO
30	REJECT		Rejects week of new measurements
31	ACCEPT		Accepts week of new measurements
32	DISTUR		Defines periods of disturbed measurements
33	SPOWER	X	Input for reduced station power
34			Phase Difference Measurements
35	† RxLORC		External Timing Loran-C
36	† RxGPS		External Timing GPS
37	† RxPORC		External Timing Portable clock
38			Phase Shifter Measurements

The preparation procedure for the SYNCIN.DAT file consists of preparing the input data in the same order as the variables in Table 2.3-1.

† External timing data set name; (x = station letter identification)

As the program now stands, much routine data that is identical from run to run must be input each time by the operator. In a typical SYNC2 run only eight of the thirty-three parameters are normally required. All thirty-three variables must be provided however. This is typically accomplished by copying and editing the previous week's input file. Furthermore, although NAMEIN requires that all values be provided, the routine initializes all variables to default values even though they will be overwritten automatically at run time. By itself this is not a serious problem; however, it reflects the general lack of standard programming practices throughout SYNC2.

2.3.4 Output Process

The final or external output of SYNC2 consists of both tabular data and plots directed to a line printer. Fatal error messages and warning messages are also directed to the printer. This data, however, is not stored by the program and useful outputs can be lost in the event of a printer failure. To avoid this situation it is desirable that the data streams be preserved as files until the operator/CLI macro deletes them.

SYNC2 plotting is done on a line printer and the visual quality of the plots is poor; a sample plot is provided in Figure 2.3-3. If the plots are to be retained in an updated SYNC2, it would be advisable to provide the option to make use of a better quality printer. The tabular outputs are of reasonable quality but some of the labels are incorrect. A recent SYNC2 run examining the processing of GPS data provided as output a plot labelled "SYNC2 Version 02" while the program version that generated the plot was clearly version 04. Although seemingly minor, such problems can cause considerable confusion when attempting to compare the performance of different versions of SYNC2.

The plotting routine itself is very cumbersome and lacks modularity. Modifying the existing routine to operate more efficiently would be a time consuming task and would provide little improvement. One alternative to improving the quality of the plots would be to make use of self contained plotting software libraries that are available.

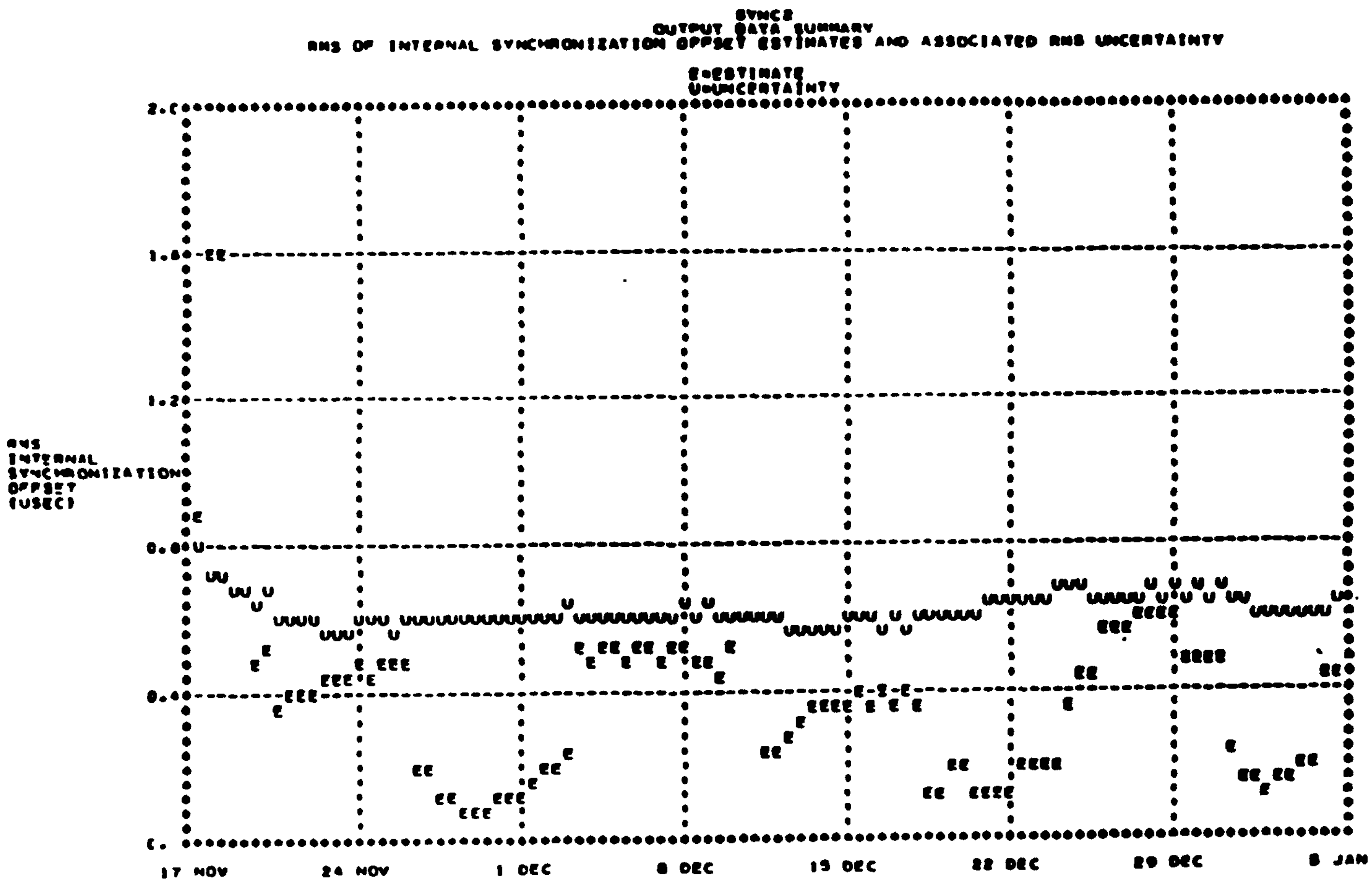

Figure 2.3-3 SYNC2 Line Printer Plot

2.3.5 User Interface

The user interface does not provide sufficient "onscreen" feedback following a program exit due to a fatal input data error. In the case of an error, an error message is sent to the printer but no message is given at the terminal. Minimally a brief error message should be provided to the terminal and ideally, all fatal error messages should be available to the terminal.

A similar situation exists regarding warning messages. Warning messages generated after the prefilter process are provided to the printer but not the operator's screen.

2.3.6 Incomplete Program Modifications

In examining SYNC2, several instances of incomplete program modifications have been noted. The examples provided below indicate the haphazard manner in which many modifications have been made leaving the integrity of the entire SYNC2 coding in question.

1) Vacant "DO" loops

    A double "DO" loop occurs in subroutine EXTRAP at label 333 without other executable statements in the area of P matrix extrapolation.

    A "DO" loop occurs in subroutine PPCIN without any other executable statements.

2) Unusable input functions

    For unknown reasons the following input functions have been modified and are presently unusable. However, coding for these functions still clutters the program.

    Predicted propagation corrections: PPC102 PPC136
    These input variables were introduced to overlay PPC data from the PPC input file if required. The data structures are still in place, and if PPC102 or PPC136 are present in the input file they are read in and stored. None of this stored data is referenced again by any SYNC2 routine.

Reduced station power: SPOWER
The purpose of the SPOWER input was to allow the program to exclude from timing computations any phase difference measurements of a transmitter whose power had dropped below a stored threshold. The program appears to have the coding for this function intact, but the power thresholds in common storage are initialized to zero for all stations. This feature should not be used without adequate testing. Since there is no existing documentation that will allow this feature to be reconstructed, it should be considered as suspect.

Set transmitter to tracking mode: TRACKT
The purpose of this input was to allow the program to accumulate phase difference data for a recently restarted transmitter while eliminating this data from current calculations. The TRACKT input invoked this "tracking" mode. This feature was in use before the SYNC2 program was ported to the Data General MV8000, but there is no record of modification to this function. However, modifications were definitely made as evidenced by a coding error in subroutine WARN. If the TRACKT input is used now for any transmitter, the SYNC2 program will exit with a Fortran fatal error (The variable name "IDAYS" in a statement in WARN has been misspelled "IDAY", resulting in a run time value of zero. The result of executing this statement will be a negative index to the variable "DATES"). This coding error shows that this feature is highly suspect even after making the obvious correction. In subroutine INIT, division by zero and subsequent fatal system error will result if all stations are in tracking mode. No screen message was sent to the operator. Since there is no existing documentation that will allow this feature to be reconstructed, it should be eliminated.

3) Other misplaced code examples:

No path to six statements in subroutine GETDAT (dead code) in the "Validation of Disturbance Data" area; no note in code with explanation.

In subroutine PHSHIN a write statement at Line # 69 uses a wrong format resulting in a Fortran system warning message trying to output fatal error message:

"****ERROR: INVALID PHASE SHIFTER DATA CARD: - - - ".

In subroutine PLOT a conditional statement on Line # 130 contains a wrong value. (Comparing "E" with index IS ranging from 1 to 4.) The result is that " (USEC) " is never printed where intended.

In subroutine PRINTZ the variable JREC is defined and is referenced on Line #384, but this variable has never been loaded. It is assumed that this results in an erroneous value of an element in the STORZ array (Assumption: JREC was used instead of JREJ).

2.4 Software Documentation

Operating and maintaining a program such as SYNC2 requires detailed software documentation. Standard software development guidelines call for two main software documents; one describing the functions and algorithms to be implemented, and one describing the details of the implementation. As part of this second document, detailed pseudocode is required along with a mapping between the actual code and functional requirements. Although SYNC2 was not explicitly developed to meet such strict guidelines, such guidelines provide a means of measuring the state of the SYNC2 documentation.

There is one final reason why the status of the SYNC2 documentation is a special concern. At the end of 1987, ONSCEN's only scientist who had started to work in the "transition" years departed. While at ONSCEN, this scientist provided the link between the original Navy synchronization scientist and the modern operators of the synchronization process. Furthermore, this individual was in charge of SYNC2 configuration control and the overall SYNC2 upgrade process. With his departure, the need for updated documentation and formal configuration control practices has become more important.

2.4.1 Functional Requirements Documentation

The functional requirements for SYNC2 are provided in two main documents. The SYNC2 version 2 release was accompanied by the "Omega Synchronization Computer Program Documentation". This document provides general description and operating instructions. SYNC2 version 3 was also accompanied by documentation describing the functional modifications between version 2 and version 3. Taken together, these two documents provide a reasonable baseline

functional software description. No functional description documentation accompanied the latest SYNC2 2, version 4, which included the addition of GPS measurement processing capabilities.

The functional software description provided by the sum of these two documents has become dated and does not accurately reflect the current program. Several items have either been added or deleted from the program without being documented. Additionally, the operating environment has undergone changes. Several example discrepancies are:

- GPS measurement processing - No description of the GPS measurement processing is provided.

- Batch mode processing - documentation describes a batch mode processing option that no longer exists.

- Omega Station Changes - documentation describes the system when Trinidad was a station transmitting in the place of the Australia station which became operational in 1982.

- Loran-C Chain realignment - The original documentation speaks only of "upcoming" Loran-C measurements to be made at North Dakota. This has been overtaken by events as the Loran-C chain used was subsequently realigned.

- Loran-C propagation Dynamics - besides changing the mean offset value, Loran-C signal dynamics have changed with the new chains. A refinement has been made to account for Loran-C propagation dynamics but has not been documented.

2.4.2 Software Implementation Documentation

Until recently, no software implementation documentation outside the actual code existed. The twenty seven routines that made up SYNC2 were poorly commented and were not (until recently) accompanied by pseudocode. This lack of documentation makes it very difficult to verify or trace the implemented algorithms. The sample subroutine given previously in Figure 2.3-1 clearly illustrates the difficulties in extracting algorithms from the code.

The recently developed "SYNC2 Programmer's Reference Manual" [Reference 3] provides a more accessible look at the SYNC2 implementation, and covers some of the requirements of a software implementation document. This effort helped bring to light many of the coding errors identified in SYNC2. Given the state of the code, it would probably be prudent to defer the development of more detailed software implementation documentation until the code is upgraded.

3. CONCLUSIONS AND RECOMMENDATIONS

The current SYNC2 program, baselined in 1976, has been a key element in the Omega synchronization process, allowing synchronization on the order of 1 microsecond. Although SYNC2 performance has generally been acceptable, changes in the Omega operating environment (e.g., advent of GPS) and improvements in estimation practices suggest the need for a SYNC2 upgrade.

The various concerns raised in Section 2 range from simple coding errors to potential numerical instabilities resulting from the SYNC2 Kalman filter implementation. In determining how these different sets of problems should be addressed, performance requirements and other factors must be considered. The most important factor in deciding how to proceed, however, is the state of the current SYNC2 software. Because of the number of internal inconsistencies, unfinished software modifications, programming errors and the use of non-standard coding practices, the current SYNC2 is unsuitable as a baseline for any significant new development.

Given these considerations, two distinct approaches to upgrading SYNC2 present themselves. The first option approaches the problem from a "fix only if broken" perspective. Under this option a SYNC2 version 5 would be created from the existing SYNC2 version 4. Major tasks to be completed under this option are:

- Resolve identified existing incomplete software modifications where feasible
- Add "Audit Trail" to all routines
- Correct existing coding errors
- Provide option to eliminate PPC corrections
- Provide software documentation describing modifications.

The major risk associated with this approach is the poor state of the current code. The concern here is that apparently benign software modifications might collide with the existing program inconsistencies and yield unpredictable results. It is conceivable that a considerable debug effort might be required under this option.

The second option approaches the problem as a complete SYNC2 redesign and would lead to an essentially new program - SYNC3. This more ambitious option would seek to significantly streamline the synchronization scheme, employ state of the art estimation algorithms and software development practices, and aim for rehosting on a PC. The existing SYNC2 algorithms, and to a significantly lesser degree the SYNC2 code, would serve as a starting point. Although requiring a greater level of effort, the end result would be a more efficient, easier to maintain, more reliable program offering an improved user interface. Major tasks to be accomplished under this option are:

- Re-examine the data processing scheme and develop a measurement quality weighting independent of current data
  - Implement Biermann's Sequential U-D algorithm
  - Employ nominal values in computing the R matrix

- Improve program Input/Output Data handling
  - Add a preprocessor of the weekly input reports to set up verify and perform data reduction where required.

- Target SYNC3 for PC environment

- Provide complete functional/software documentation.

This second approach, unlike the first, has few hidden technical risks, but calls for a greater level of effort to implement. In establishing a SYNC3 baseline, a more structured approach to software development and documentation must be followed. Within this approach, the appropriate software development standards must be maintained. Targeting SYNC3 for a PC class machine would not preclude the option to run SYNC3 on its current platform.

APPENDIX A EXISTING SYNC2 SOFTWARE DEFICIENCIES

1. CODING ERRORS:

Routine COMPU:

The variable NQ (clock status change) changes from zero to one, but is never accessed.

The variable AFREQ is defined and initialized, but is never accessed.

Routine EXTRAP:

A double loop appears at label 333 without other executable statements.

Routine FILTER:

The variable MM is loaded but is not referenced.

Routine FILTIN:

The local variables LOOKI and LOOKII are calculated, but never accessed.

Routine INIT:

A division by zero and fatal system error occurs without an error message if all stations are tracking.

Routine MTSMPL:

A spurious variable "DETM" is generated by the compiler due to an "M" in column 72 of source code. This would have resulted in fatal system error. Division by zero occurs if determinant of working array W2 not singular. (Error fixed at ONSCEN on 12/14/89.)

Routine NAMEIN:

The variables SIGFSQ, SIGFD2, ADJLIM show useless initialization. Following initialization by data statements, the same variables are overlayed by data that is automatically read in.

The variable DMISS7 in /FILT/ common is defined and initialized, but is never accessed.

The variables PPC102 and PPC136 in blank common are defined and data is read in, but it is never referenced.

Routine PHSHIN:

The write statement at Line # 69 uses a bad format resulting in a Fortran system warning message trying to output fatal error message: "****ERROR: INVALID PHASE SHIFTER DATA CARD: - - - ". Format 1051, Line # 70 is wrong.

Routine PLOT:

The formats 1021, 1022, 1023, and 1024 are defined, but never used.

The conditional statement on line # 130 contains wrong value. (Comparing "E" with index IS ranging from 1 to 4.) The result is that " (USEC) " is never printed where intended.

Routine PPCIN:

If conversion error occurs for A2 -> I2 conversion, then the program exits without any message. Bad loop at Line # 182: No executable statements in loop.

Routine PRINTZ:

The variable JREC is defined and is referenced on Line #384, but the variable has never been loaded. It is assumed that this results in erroneous value of variable STORZ element. (Assumption: JREC was used instead of JREJ.)

Routine SAVE:

The variables LOOKI and LOOKII are defined and initialized, but never accessed.

Routine WARN:

An index variable IDAY is erroneously used in a statement on Line # 195 instead of IDAYS. Since the IDAY variable is not initialized or given a value, this statement will result in a negative index to the DATES variable and a subsequent error exit. This is a conditional test that is only true if any station is in the "tracking" mode.

Global Common overlays:

The commons show a dangerous level of overlays within the same contigous storage area, namely:

Common name:	Number overlays:
BLNK	1
BUFF	5
CLK	1
FILT	2
NEW	11
ON	5
OUT	4

The common named "NEW" exhibits dangerous variable naming convention by different routines, namely:

REJCT(30,2) is overlayed by REJECT(30,2) and
REJECT(9,8,2) is overlayed by REJCT(9,8,2).

System I/O Errors:

With the exception of testing for an open error for the SYNCIN.DAT file, no I/O errors are trapped with the result that in case of an I/O error, the program exits without the ability to record the cause.

Fortran Arithmetic Errors:

The program does not attempt to trap arithmetic errors as over/underflow (and division by zero) to aid with recovery procedures.

2. COMPILATION WARNINGS:

All routines using Common /ON/:

- This common is defined with CHARACTER*1 TRAN(10) variable in the middle of variables of types REAL*4, INTEGER*4.

Routines COMPZ,INIT,NEWH:

- DO-loop index in routine call-list.

Routine GETDAT:

- Six statements of dead-code in "Validation of Disturbance Fields" area (no path to code). No note in code with explanation.

3. LANGUAGE USAGE, ETC:

Computational accuracy:

- Too much use is made of mixed mode computations.
- No use made of Real*8 variable typing for critical variables.

Error-Risks:

- Modules make use of Blank-common.
- Extensive use made of Default Implicit variable typing.
- No mechanism used to force all named commons with same label to have same contents.
- Mixing character and numeric data in common.
- Logical variables used as integers.

Maintainability:

- No use made of indentation of code to improve readability.
- No permanent set of debug statements included.
- No permanent record in routine source code of modifications to program, time and date of modification, and by whom.

- Line-ID starting in column 72 of most statements does not serve any purpose after conversion from card input. This is definitely a hazard when making changes to program.

- Too few remarks and comments to accompany code.

Operability of software:

- No exit messages displayed at terminal.

- Requires input of rarely used data for every run.

- Seasonal changes of data requires program change and rebuild process.

- Operator must get warning messages off printer before he can determine a GO/NOGO to continue execution.

Run-time Efficiency:

- 90% + of coding is written in Fortran IV; this is very inefficient code for present Fortran 77 System.

- Text/ASCII format is used for internal files.

4. INCORRECT OR QUESTIONABLE ALGORITHMS:

Routine FILTIN:

Average frequency offset is calculated based only upon on-air transmitting stations; however, this average is subtracted from every station's frequency offset estimate.

Routine GETDAT:

The capability to use SPOWER variable to delete measurements taken during low power operation has been lost by setting the power-threshold (local!) matrix AMPTHR = 0 for all stations. (Question: What constitutes a "valid" power-threshold for obtaining acceptable measurements - 80,90% ?)

Portable Clock Input:

Two separate inputs are required to supply the program with portable clock data: variable MESTIM for GMT of clock measurement time, and an RxPORC input line for the external timing measurement.

Scaling of plots:

Variable scaling for plot Y-axis is only available for four plots. (FESCAL, FUSCAL, SESCAL, SUSCAL)

APPENDIX B

This appendix presents the results of simulation runs performed to examine reported SYNC2 anomalies. When external GPS measurements are removed from a station, it has been reported [References 1 and 2] that SYNC2 attempts to drive the station back to its internally referenced time. This time can often be two or more microseconds from its externally referenced state.

The simulation scenarios examined considered three stations, a full set of internal measurements, and a varying number of external GPS measurements. In both of the two cases discussed below, stations "X" and "Y" have GPS measurements throughout the 90 day interval. Station "Z" is assumed to have GPS measurements only for days 30 to 60. In both cases, a complete set of internal measurements is available throughout the interval. These measurements processed in an eight-state Kalman filter designed to simulate the basic processing performed by the SYNC2 filter.

In the first case, the internal and external measurements were assumed to be mutually consistent. Given the consistent measurement set, the filter's estimates remain stable when the external timing measurements are removed from the station. Figure B-1 shows the three internal phase offset estimates for each station and the external system phase offset estimate (UTC relative to Mean Omega System Time). Figure B-2 shows the external phase offset for each station computed as the difference between the external system phase offset and the respective internal phase offsets.

In the second case, an inconsistency between the internal and external measurements is introduced. Specifically, one of the internal measurements associated with station "Z" is assumed to be "in error" by 3 microseconds. Figure B-3 shows the filter estimates of the internal phase offsets and external system offset for this test case. Figure B-4 shows the external phase offsets for each of the three stations. As expected, a large shift in the estimates occurs at day 30 when GPS measurements become available at station "Z". When the GPS measurement associated with station "Z" is removed at day 60, the stations drift back to their former values.

As these test cases illustrate, inconsistencies between the internal and external measurements may give rise to the type anomalous behavior reported when external measurements are removed. Given a consistent measurement set, the simulation shows that the estimates remain stable.

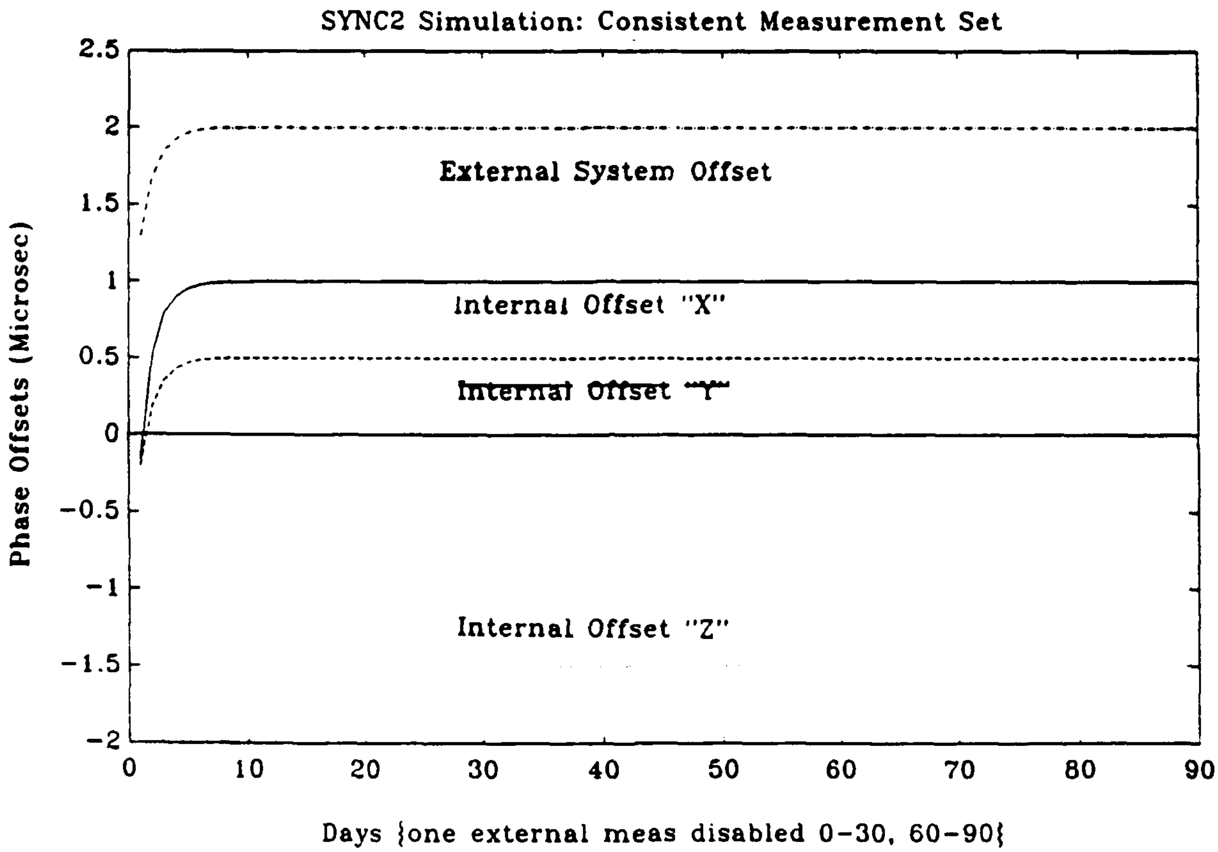

Figure B-1 Consistent Measurement Set: Internal and System Offsets

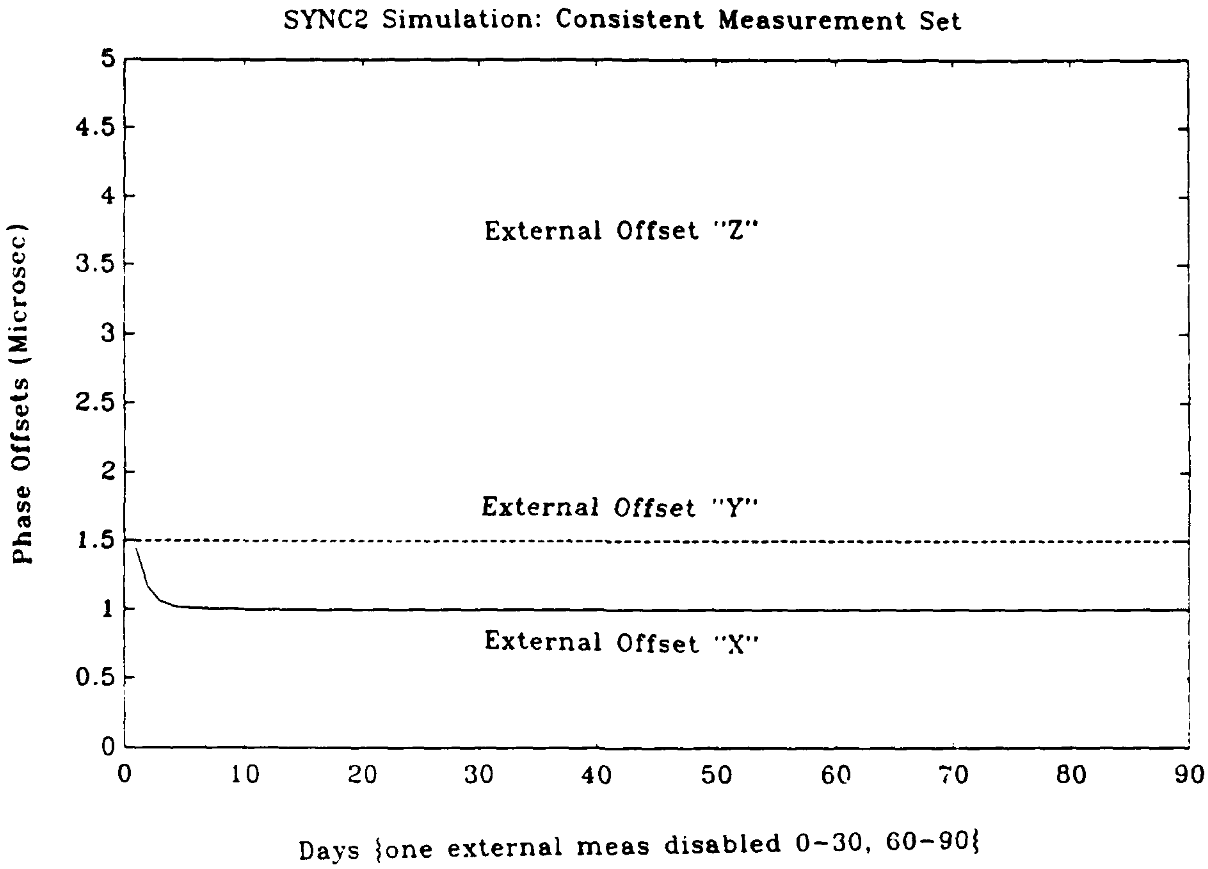

Figure B-2 Consistent Measurement Set: External Phase Offsets

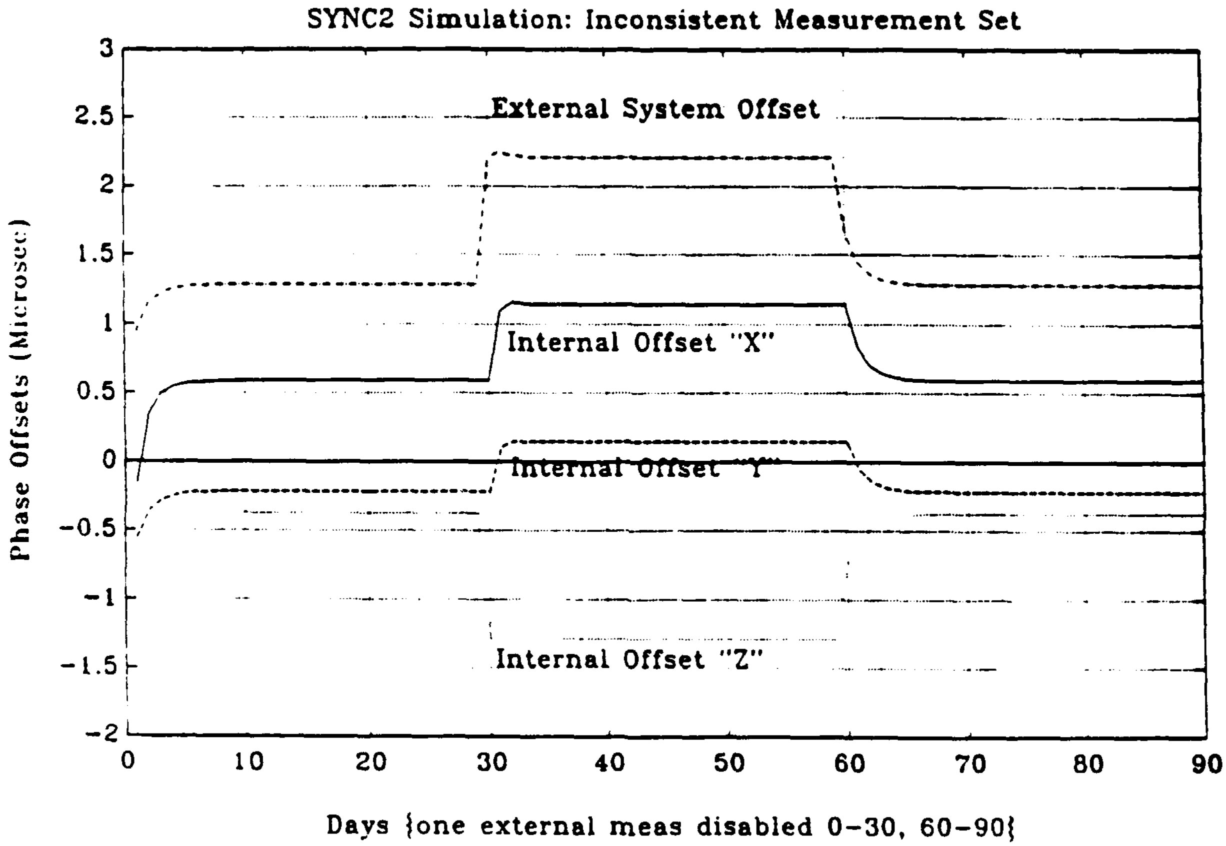

Figure B-3 Inconsistent Measurement Set: Internal and System Offsets

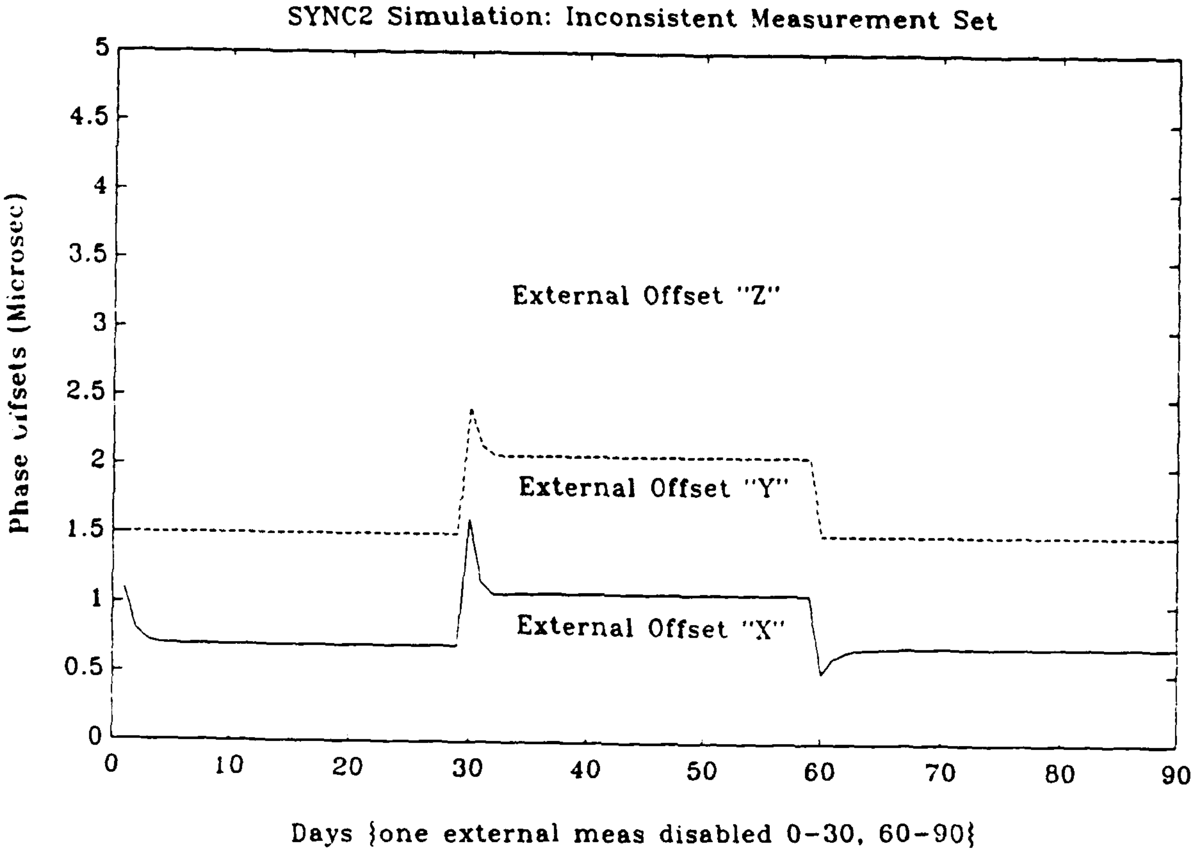

Figure B-4 Inconsistent Measurement Set: External Phase Offsets

REFERENCES

1. LT. R.C. Thomson, "Synchronization of the Omega Navigation System Using External Timing References", Proceeding of the Fourteenth Annual Meeting - Internaional Omega Association, 6 October, 1989.

2. LT. R.C. Thomson, "An Overview of Omega Synchronization", April, 1990.

3. *SYNETICS*, "SYNC2 Programmer's Reference Manual", April 18, 1990.

4. *SYNETICS*, "SYNC2 Operator's Manual", June, 1990.

5. The Analytic Sciences Corporation, "Omega and Synchronization Computer Program Documentation", 30 January 1976.

6. The Analytic Sciences Corporation, "SYNC2 Program Modifications", 30 November 1979.

7. *SYNETICS*, "Analysis of GPS Timing Data In Support of Omega System Synchronization: A Cesium Stability Study", 21 April 1990.

8. Bieman, Gerald J. "Factorization Methods for Discrete Sequential Estimation", Academic Press, Inc., London.